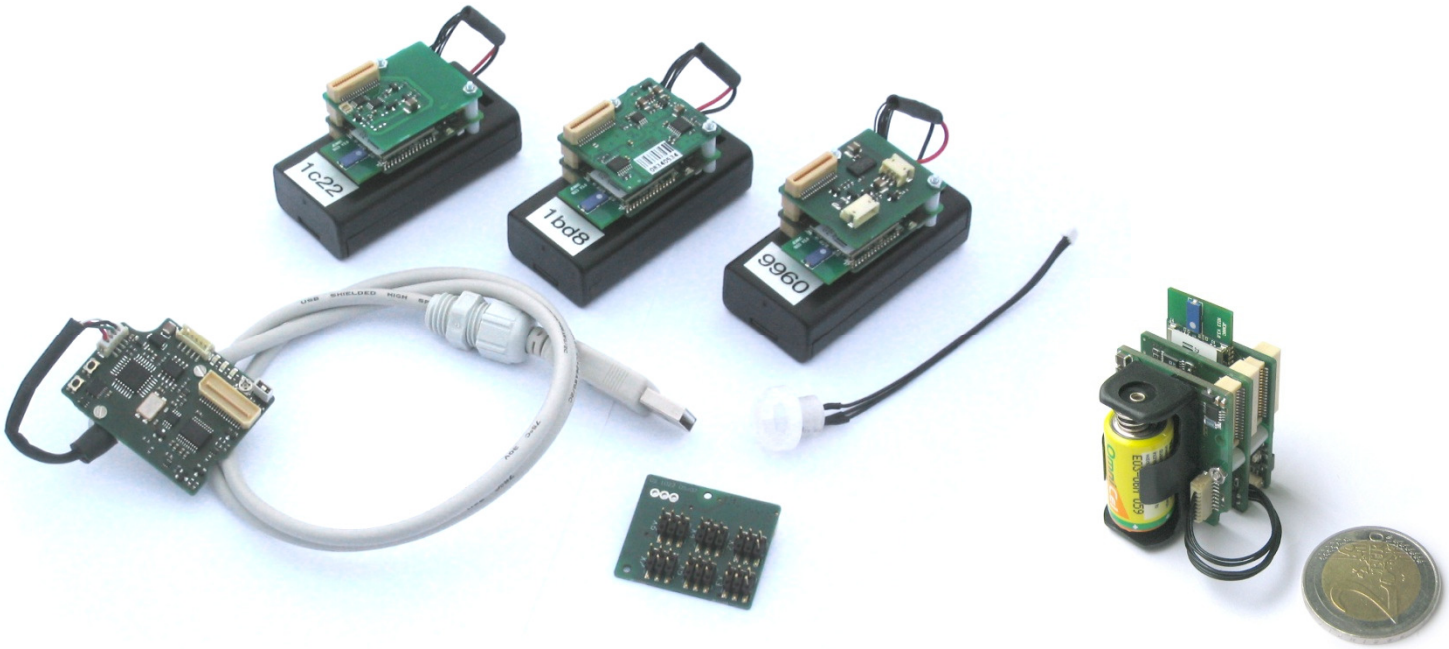


Flashing iSense Devices Wirelessly User Guide

coalesenses
research to innovate

Coalesenses GmbH Maria-Goeppert-Str. 1 23562 Lübeck Germany www.coalesenses.com sales@coalesenses.com

Document history

Version	*Date*	*Changes*
1.0	15.10.2008	Initial version
1.1	08.12.2008	Adaptation to changes in improved over the air programming
1.2	31.10.2009	New Layout
1.3	04.06.2010	Adaptations to the changes in the Over-the-Air-Programming Plugin for iShell

Contents

1. About this User Guide 4
2. Making Applications support OTAP 5
3. Connecting a gateway node to the PC 7
4. Using the Over-the-air Programming plugin in iShell 10
5. References 12

1. About this User Guide

This guide shows how to program iSense devices wirelessly. This process is also knows as Over-The-Air-Programming (OTAP). Note that this guide only refers to 1 hop wireless programming, as the already available multi-hop wireless programming support is still experimental.

In this user guide,

- files and folders are represented in the Arial typeface,
- code fragments, function names etc. are represented in the `Courier New` typeface,
- GUI elements such as button descriptions etc. are represented in “quotation marks”,
- titles of other documents are presented in *Italic* type.

This manual assumes that the reader has successfully installed the iSense development environment, and obtained either the iSense standard firmware or a custom firmware from the iSense firmware web configuration interface. For further information on these steps, consult the *Development Environment Setup User Guide* [1].

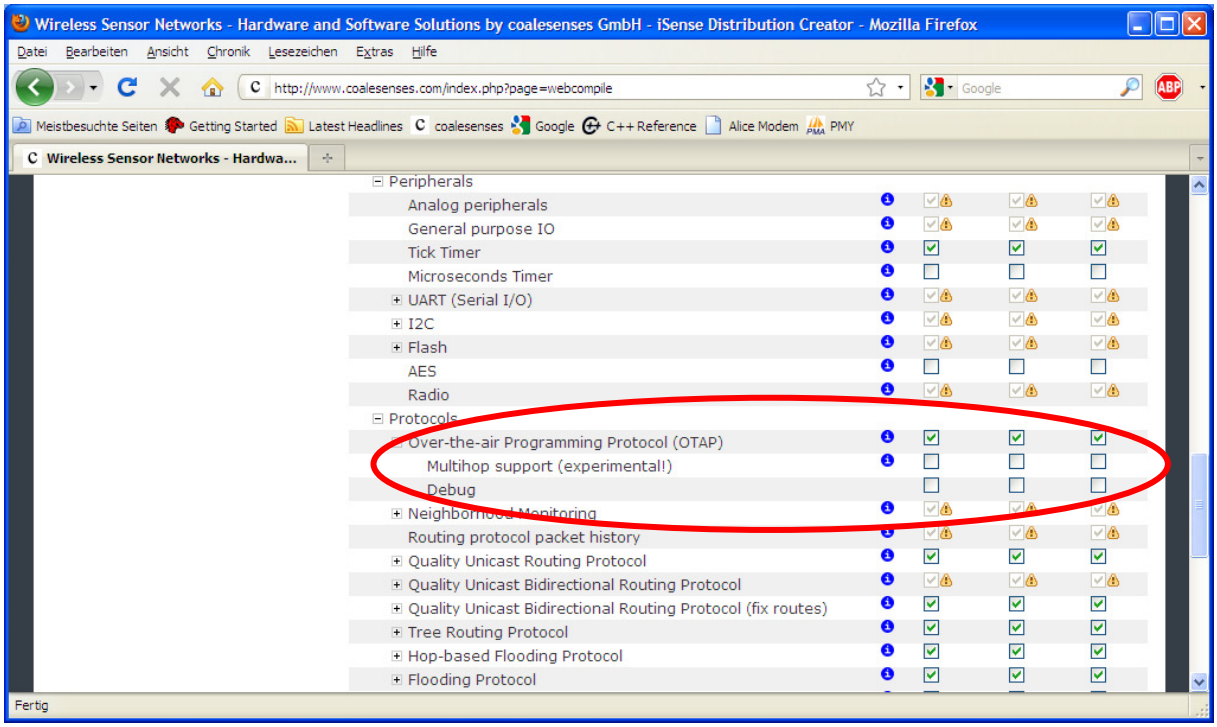
In addition, it is assumed that the user is familiar with the use of iShell. For further information on iShell, consult the *iShell User Guide* [2].

For further information on iSense firmware programming concepts and on application development, it is recommended to read the *Writing iSense Applications User Guide* [3].

2. Making Applications support OTAP

Before devices can be flashed wirelessly, they must once be flashed by wire with an application that supports later wireless programming.

To make your application support OTAP, all you have to do is compile and link your application against a firmware that supports OTAP. To obtain a firmware supporting OTAP, either use the standard firmware, or enable the OTAP feature in the configuration of firmware you obtain from the *iSense firmware web configuration interface* [5]. To do so, go to the protocols section, and enable "Over-the-air Programming Protocol".



Details on how to use the *iSense firmware web configuration interface* can be found in the *Development Environment Setup User Guide* [1].

While stable OTAP protocol only supports single-hop OTAP, there is also an experimental multi-hop version. However, its use is currently not recommended in production environments. To active multi-hop OTAP support anyway, also select "Multihop support (experimental)" on the *iSense firmware web configuration interface* [5].

Now compile and link your application against the resulting firmware, and it is OTAP-ready.

However, a few issues should be kept in mind.

If you run a larger number of devices with potentially different applications, and you want to flash all of them wirelessly, it might be difficult to identify which device runs which application. To alleviate this difficulty, the class `isense::Application` (that all applications must inherit from) provides two methods for returning an application id as well as an application revision.

```
//------------------------------------------------------------------------------
/** This method is used to identify an application.
 * If you program devices over the air it eases the selection of the desired
```

```
     * devices.
     *
     */
    virtual uint16 application_id (void);
    //------------------------------------------------------------------------
    /** This method can be used to differentiate between different versions of
     * the application. The number is also (in addition to the application id)
     * shown in the OTAP-Plugin, so that the desired devices can be selected.
     *
     */
    virtual uint8 software_revision (void);
```

This information is taken up by the OTAP protocol, and will be displayed on the iShell user interface. Hence, your application is recommend to implement these methods and return characteristic values here. In case you do not implement (i.e. overwrite) the above functions, their default implementations (returning 0 for both) will be used.

You should also keep in mind that you will only be able to flash you devices wirelessly if they wake up and have the radio turned on every once in a while. As a rule of thumb, devices should remain awake and listening for at least 300ms to be catchable by the OTAP protocol. Once your devices noticed that an OTAP session is going on, they will stay awake automatically if they are to be flashed wirelessly, and will go back to their particular sleep and wake cycle in case they are not selected for reprogramming. For details on how to control whether devices are awake or not, please refer to the *Writing iSense Applications User Guide* [3].

You can additionally protect your devices from unauthorized wireless reprogramming by setting a 128bit AES key:

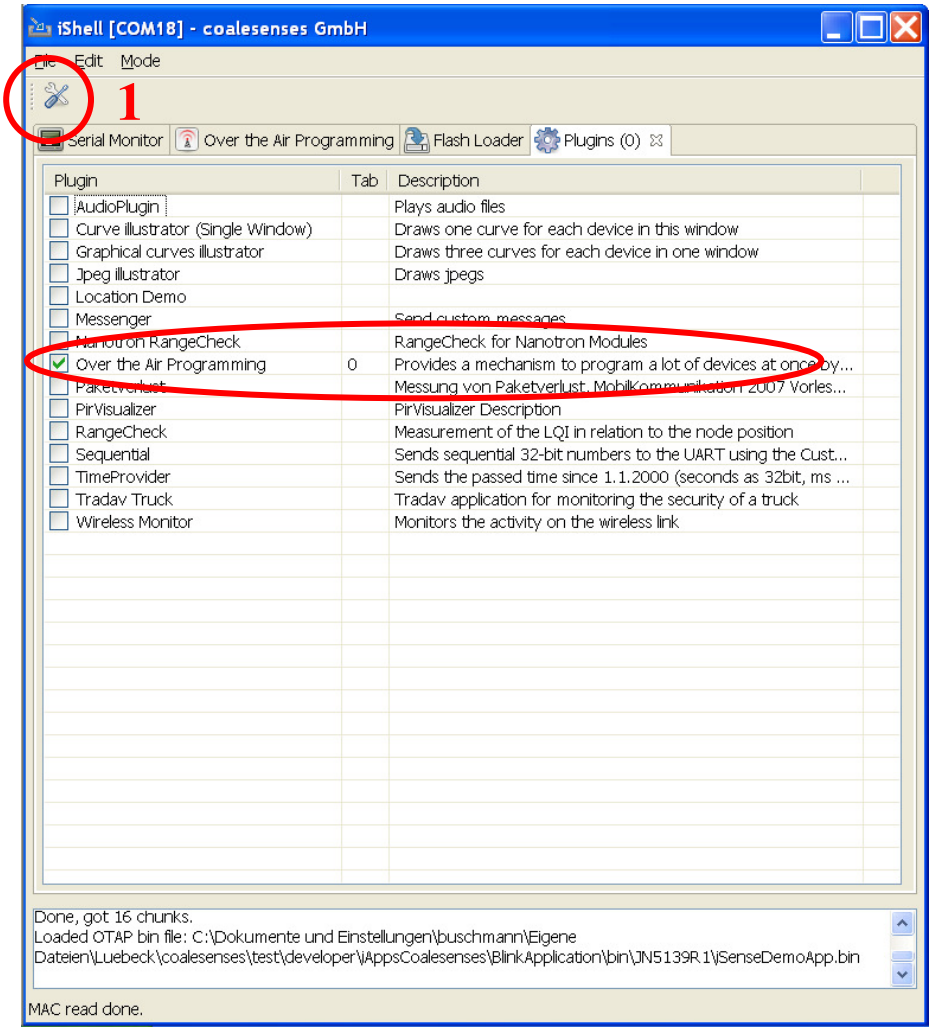
```
// define a sample key
uint8 key[16] = { 0x0, 0x1, 0x2, 0x3, 0x4, 0x5, 0x6, 0x7,
                  0x8, 0x9, 0xA, 0xB, 0xC, 0xD, 0xE, 0xF };
bool result = os().otap().enable_security(true, key);
if (result == true)
       os().debug("OTAP key set successfully");
else
       os().debug("Failed to set otap key.");
```

Devices with a set OTAP key will not be detectable by the OTAP protocol if it is run without a key or with a different key.

3. Connecting a gateway node to the PC

Make sure you are using the latest version of iShell.

Start iShell. In case the "Over the air programming" tab is not visible, go to the "Plugins" tab, and activate the "Over the air programming" plugin by selecting the checkbox on the left.



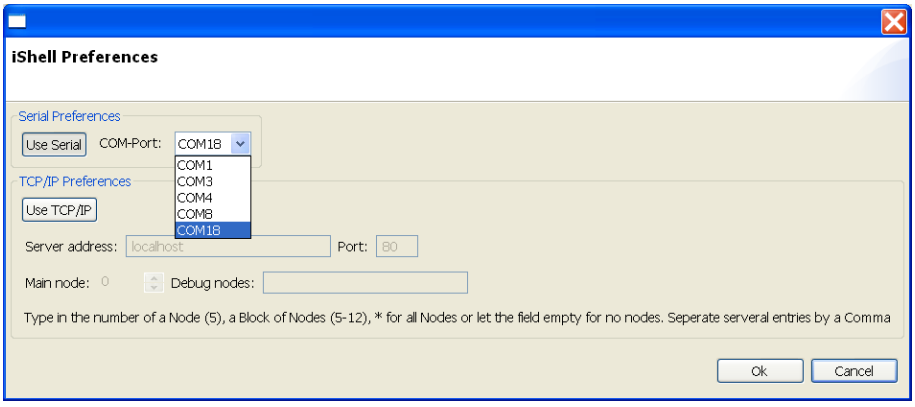
Connect an iSense device to the PC to act as a gateway between the PC and the devices to be flashed wirelessly. It must at least consist of

- an iSense Core Module and
- an iSense Gateway Module connected to the PC via a USB cable,

or

- an iSense Core Module,
- an iSense Gateway Module connected to the PC via a serial cable, and
- an iSense Power Module.

Select the port the device is connected to in iShell. To do so, click on the preferences button (1).



The "iShell Preferences" dialog will open. Choose "Use Serial", select the correct COM port, and click on "Ok".

Make sure the device connected to the PC is flashed with the iSerAerial application.

You can download this application from the coalesenses web site. To do so, go to www.coalesenses.com, and click on "Downloads", then click on "iSense Applications" and download the iSerAerial application binary. Be sure to choose the binary file version that fits your gateway device type:

Core Module	*Controller type*	*Addressing mode*	*Binary*
CM10X	JN5139R	16 bit short addressing	iSerAerial_JN5139R0_16bit.bin
	JN5139R	64 bit extended addressing	iSerAerial_JN5139R0_64bit.bin
	JN5139R1	16 bit short addressing	iSerAerial_JN5139R1_16bit.bin
	JN5139R1	64 bit extended addressing	iSerAerial_JN5139R1_64bit.bin
CM20X	JN5148	16 bit short addressing	iSerAerial_JN5148_16bit.bin
	JN5148	64 bit extended addressing	iSerAerial_JN5148_64bit.bin

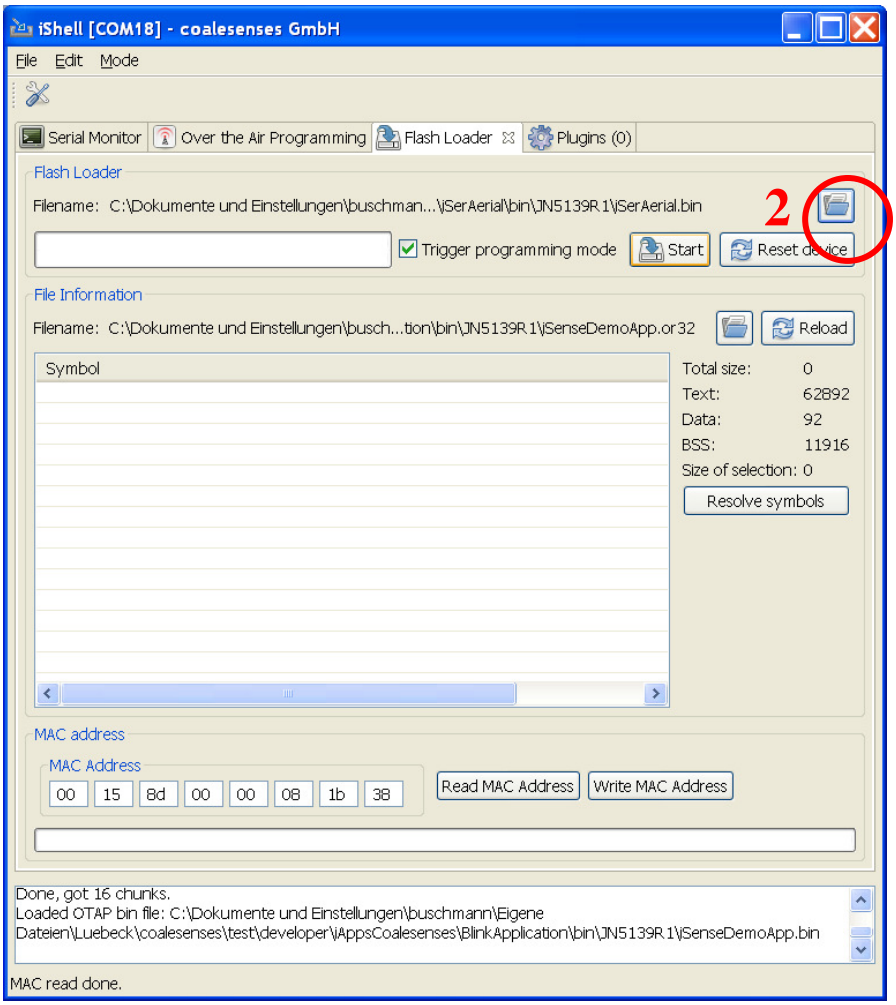
In the common case, owners of a first generation Core Module (CM10X) will find the JN5139R1 controller on their module. For details on how to identify the controller type on your first generation Core Module, please refer to the *Core Module User Guide* [4].

The addressing mode is not dependent on the hardware, but on the configuration of the firmware you use. It can be selected using the *iSense firmware web configuration interface*. By default, the addressing mode is set to 16 bit short addressing (as well as in older firmware revisions where no choice was possible).

Note that for wireless flashing of devices using 16 bit short addressing, you will need a gateway node with the iSerAerial application that also uses 16 short addressing (ditto for 64 bit extended addressing).

However, it is not required that the device types of the gateway and the devices to flash are the same.

Now change to the "Flash Loader" tab in iShell, click on the file select button (2), and choose the iSerAerial binary downloaded from the coalesenses web site. Make sure you are using version 1.1 or newer.

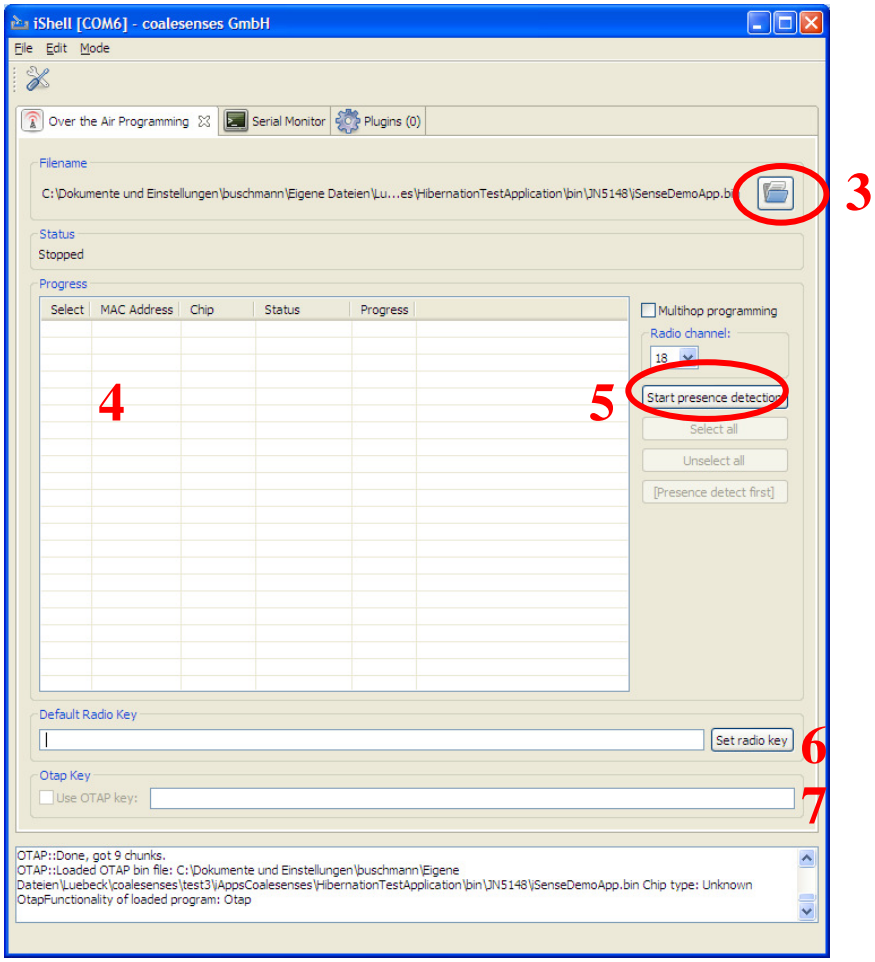


Flash the binary to the iSense device connected to the PC. To do so, choose “Trigger programming mode”, and click on “Start”, and wait for program download to complete.

Make sure that all devices that are to be programmed wirelessly are programmed with an application that was compiled with the “Over-the-air Programming Protocol (OTAP)” feature activated in the iSense operating and networking firmware, and that the firmware is newer than revision 2768.

4. Using the Over-the-air Programming plugin in iShell

Change to the "Over the Air Programming" tab in iShell.



Click on the file selector button (3), and choose the binary file to be transferred to the devices. Be sure to choose a file that was compiled for the correct chip revision (in most cases that will be JN5139R1 for CM10X devices, and JN5148 for CM20X devices), and that all devices that you want to program wirelessly feature the same chip revision. For details on how to determine your chip revision, please refer to the *Core Module User Guide* [4].

Now choose the channel on which the devices you want to flash communicate. By default, devices use channel 18.

In case your devices use radio encryption (note that radio encryption is only available when using 64 bit extended addressing, and must be manually enabled by calling `os().radio().hardware_radio().set_security(true, key);`), enter the appropriate key into the "Default Radio Key" input. The key must be input in hexadecimal format, without any deviding characters, and must have a length of 16 bytes (i.e. 32 digits). Then click on "Set radio rey" (6). To unset the radio key, completely delete the key, and click on "Set radio key"

In case your devices use an OTAP key to protect themselves from unauthorized wireless reprogramming, enter the according key into the "OTAP key" input (7), and select the "Use OTAP key" checkbox. For setting the key from the OTAP key example in section 2, enter "000102030405060708090A0B0C0D0E0F". Note that using an incorrect radio key and/or OTAP key will result in failure to detect the according devices.

To detect the presence of the devices to be programmed wirelessly, click on “Start presence detection”. Nearby devices will then show up in the device list (4) if their program was compiled with the OTAP feature enabled. Note that in order to be detectable, devices must have wake periods of at least 300ms. For details on how to control whether devices are awake or not, please refer to the *Writing iSense Applications User Guide* [3].

The application ID and the application revision displayed in the “Status” column are the ones that the functions

```
//------------------------------------------------------------------------------
/** This method is used to identify an application.
 * If you program devices over the air it eases the selection of the desired
 * devices.
 *
 */
virtual uint16 application_id (void);
//------------------------------------------------------------------------------
/** This method can be used to differentiate between different versions of
 * the application. The number is also (in addition to the application id)
 * shown in the OTAP-Plugin, so that the desired devices can be selected.
 *
 */
virtual uint8 software_revision (void);
```

of your applications on the devices return. In case you did not implement (i.e. overwrite) the above functions, their default values will be displayed (App. 0 rev. 0).

Devices that cannot be programmed wirelessly because

- the actual chip type and the chip type the software was build for do not match or
- their OTAP version is too old

will be displayed with grey font color, and will be excluded from wireless flashing.

Select the devices you want to program wirelessly by checking the corresponding checkbox left of the device address in the device list. You can stop the presence detection by clicking on “Stop presence detection”.

Note that the wireless programming only work within the direct neighborhood of the device that is connected to the PC. Currently, multi-hop programming is not supported.

You can program up to 50 devices in one run.

The “Start” button (5) will become active if presence detection is running. After choosing the devices to be programmed, click on “Start”. The programming process will start. The “Status” column and the “Progress” column in the device list will inform you about the state of the different devices.

You can stop the programming process by clicking on “Stop” (the former “Start” button, 5). Devices that did not complete the reprogramming will remain functional. However, their program flash memory will be corrupted. Hence, they will not work properly after reboot or after being switched off and on again.

After the programming completed, the devices that received the complete new program will reboot automatically, and run the new program from now on. Devices that failed to receive the complete program will not reboot, and keep running the old program. However, their program flash memory will be corrupted. Hence, they will not work properly after reboot or after being switched off and on again, either.

For devices that did not complete the program download, you can repeat the wireless programming procedure.

5. References

[1] coalesenses Development Environment Setup User Guide, online available at http://www.coalesenses.com/index.php?page=development-environent

[2] coalesenses iShell User Guide, online available at http://www.coalesenses.com/index.php?page=development-environent

[3] coalesenses Writing iSense Applications User Guide, online available at http://www.coalesenses.com/index.php?page=isense-applications

[4] coalesenses iSense Core Module User Guide, online available at http://www.coalesenses.com/index.php?page=hardware-docs-and-demos

[5] coalesenses iSense firmware web configuration interface, online available at http://www.coalesenses.com/index.php?page=isense-firmware

coalesenses GmbH
Maria-Goeppert-Str. 1
23562 Lübeck
Germany

www.coalesenses.com
sales@coalesenses.com